

# **Google Assistant-controlled IoT-based Door Security Alarm**

**J Component Project Report for the course**

**Human Computer Interaction - CSE4015**

by

**Aditya Kumar Gupta - 20BPS1030**

**Ankit Kumar Jha - 20BPS1050**

**Mugdha Kondhare - 20BPS1095**

Submitted to –

**Dr. Ganala Santoshi**



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**VELLORE INSTITUTE OF TECHNOLOGY**

**CHENNAI - 600127**

## **ABSTRACT: -**

Traditional locks require a physical key to take a lock and unlock action. Therefore someone needs to be present in order to insert the key into the lock and physically turn the key. This process has been the same since locks were first invented. Smart door locks are more suited to today's world, where flexibility is cherished. The fundamentals of a smart lock are still the same as with a traditional lock - the smart door lock is able to lock and unlock a door. The major difference is that it also includes a 'smart' component so the lock can be controlled from your phone, through smart technology or an integrated smart system. Nowadays, the most common issue with door locking systems is security. Anyone with a hard object can break the door and rob the home, office, or any other property. This can result in massive losses for the human economy. The project intends to apply the concept of Human Computer Interaction to IoT. Here, we've installed a door security alarm system. Google Assistant is in charge of this door security alarm system. The system allows for remote door access and voice alerting via smart phone, as well as sending an email alert. In today's world, a smart home security control system is unavoidable. Smart door locks enable you to interact with your lock in a more flexible way, thereby giving you and those responsible for the lock far greater control and convenience of who can lock and unlock it. It's possible to adapt your existing locks and in so doing, bring them onto a platform which makes them smart. This would also include electronic locks (e.g. maglock) which haven't been designed to be used in a smart fashion - however, when adding it to a system then they would become smart. locks that are specifically designed to be smart locks and controlled electronically, e.g. through an app or some other form of electronic interface (e.g. keycode, keypad, etc.). The beauty of adapted smart door locks is that they look the same as your existing locks - they are after all, just your existing locks but with a 'smart' component added to them to make them smarter. Native smart door locks usually look a lot larger than a traditional lock and often include an input mechanism such as a keypad.

## **INTRODUCTION: -**

An effective embedded access control system with low power consumption and costs for A wide range of commercial and security applications require smart home security and remote monitoring. Smart home security control systems are becoming more and more popular in various nations. Today, the majority of the appliances in our homes and places of business have microprocessors. Although each of these appliances has some sort of user interface, many consumers find it challenging to use the complicated features of their appliances. We have created a framework that lets users use a different user interface device that they are already carrying to communicate with appliances. Because they are widely available, have

communication capabilities that enable connection to appliances, and are already used for a variety of applications, smart phones are excellent candidates for providing interfaces. Our framework consists of a two-way communication protocol, an abstract specification language for describing appliances, and automatic interface generation software that enables user interfaces to be tailored to users and the devices they are using. Accurately identifying intruders who attempt to break in is the most crucial component of any home security system. The most natural way to provide security is to detect burglars at the door and alert the user via mobile phone. An entrance guard can be managed remotely. Home automation system is a computerized, intelligent network of electronic devices, designed to monitor and control the home appliances and lighting systems in a building. It allows users to remotely monitor and control consumer electronics through the external network such as Internet. Home automation is the emerging field that has attracted the attention in both the commercial and research field. Although wired home networks were famous at the early developments of home automation systems, nowadays wireless communication is replacing the wired system which are very messy and also difficult to setup. As per the statistical search of burglary cases happened in India rate of crime is increasing day by day. In 2013 there is 10 percent increment in burglary cases. In 2012 there is the 8.2 percent crime is due to the home theft so there is a need to develop some smart door lock system to save our home from such crime. 9 out of 10 burglars avoid home with alarm system. They would not attack the home try drop the attempt. These facts encourage us to develop our smart system which help us to provide security to residential and commercial application. There are various ways proposed and implemented for providing security to home and commercial appliances using sensors deployment ,various transmitters ,internet and these methods are commercially adopted so popularly. Potential vulnerabilities in smart locks do create security concerns. Security advisors recommend that smart locks should not necessarily be thought of as more secure than a conventional lock and key and recommend that users think carefully about settings and options.

## **EXISTING SYSTEMS: -**

The conventional key and lock mechanism is probably the security system that is employed in homes the most. It is one of the most user-friendly in terms of simplicity, but compared to contemporary picking techniques, the technology is quite dated and lacking in security. Home appliances are controlled by Bluetooth in the current system using an 8051 microprocessor. In order to turn ON and OFF any nearby electronic device, Bluetooth must have a range of 10 to 15 metres.

## **PROBLEM STATEMENT: -**

A 2017 Home burglary statistics report claims that 3 out of 4 US homes will be broken into within the next 20 years and that there is a home burglary every 13 seconds. Burglary shouldn't be a big concern in a world where technology is developing quickly to the point where cars may be able to drive themselves and drones may be able to capture your food, but the statistics above show that this isn't the case. If I can't feel safe at home, what use are all the buzzwords like IoT, AI, machine learning, etc.?

## **INTRODUCTION TO PROPOSED MODEL: -**

The overall goal of the Python script is to determine whether or not the alarm is activated. If it is, we must then determine whether the door has been opened, and if it has, we must then send an email and turn on the buzzer. In order to authenticate users attempting to access the door, our project aims to achieve a proper policy. It assesses an appropriate set to create a smart door lock that is meant to provide high security, simple access, and control.

## **PROPOSED SOLUTION: -**

Therefore, for this project, we will create a security system that can recognise when a window or door is opened. When an intrusion is detected, the alarm will also send a mail to the owner warning them about it. The alarm can be activated or deactivated using voice commands through Google Assistant. The cool thing about it is that it can be controlled from anywhere in the world because the entire thing is cloud-based.

## **COMPONENTS OF THE MODEL: -**

### **Software components -**

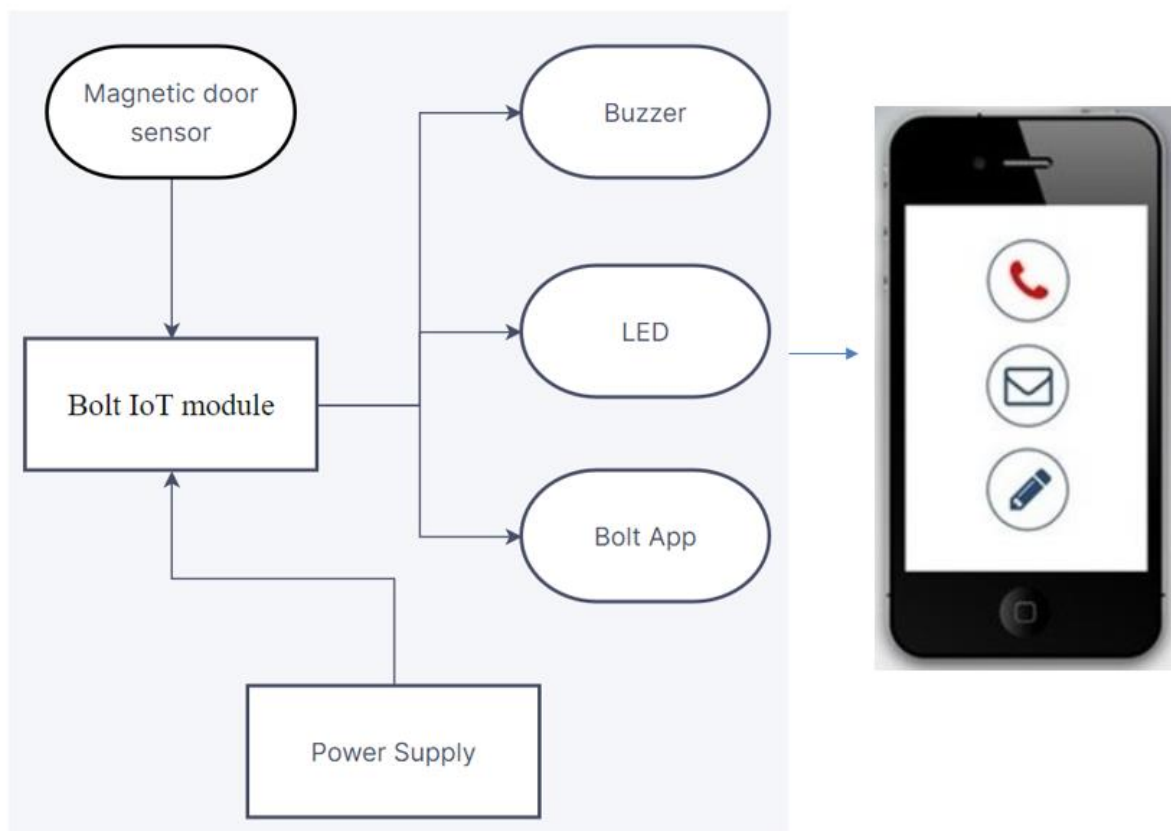
1. Bolt Cloud API - <https://cloud.boltiot.com/>
2. python3, with pip packages - <https://www.python.org/>
3. IFTTT applets - <https://ifttt.com/>

### **Hardware components –**

1. Bolt Development Board
2. Breadboard

3. Hall effect sensor (A3144)
4. LED
5. Capacitor (50V, 10uF)
6. Resistor 10K
7. Magnet
8. Connecting wires

### **BLOCK DIAGRAM: -**



*Figure 1 Block Diagram*

This is a security system that can recognise when a window or door is opened. When an intrusion is detected, the alarm will also send a mail to the owner warning them about it. The alarm can be activated or deactivated using voice commands through Google Assistant. The cool thing about it is that it can be controlled from anywhere in the world because the entire thing is cloud-based

## **DESCRIPTION OF THE COMPONENTS (HW/SW/TECH. SPECIFICATIONS)**

- 1) Bolt Cloud API- ESP8266 with custom firmware , 32-bit RISC CPU: Tensilica Xtensa LX106 , 5V/1A DC via Micro-USB port or 5V and GND pins , 64 KB of instruction RAM; 96KB of data RAM , 5 Digital pins [3.3V logic.
- 2) IFTTT applets:- Connects two or more apps or devices together. It's an automation or integration/connection between two services that enables you to do something that those services couldn't do on their own.
- 3) Bolt Development Board:- Easy interface to quickly connect your hardware to cloud.
- 4) Breadboard:- Used for building circuits.
- 5) Hall effect sensor (A3144):- Used to detect the presence of magnetic field.
- 6) Buzzer- Will start ringing when an intrusion is detected.
- 7) Capacitors:- Used as memory elements.

## **WORKING MECHANISM: -**

Based on the well-known ESP8266 Wi-Fi module from Espressif Semiconductor, the Bolt development board was created. It runs its own Bolt firmware, which enables us to use an API provided by Bolt to access the GPIO pins (Digital Read/Write, Analog Read, PWM Write, etc.). As a result, the Bolt can be programmed using Python, HTML, or even JavaScript. Python is the language we used for the programming. All 5 GPIO pins and 1 Analog pin on the Bolt are wired to the cloud. We must therefore use the API calls in order to write to or read from these pins. We have a magnet mounted to the door and a Hall Effect sensor as part of our hardware. The magnet will move away from the hall sensor when the door is opened, which will be picked up by the sensor and read by the API calls, allowing us to activate a buzzer. The Hall sensor serves as the input pin that informs the board whether or not the door is open. The LED attached to pin 2 is used to display the system status. The LED will glow if it is activated, and it will be off if it is not. When there has been an intrusion, the buzzer goes off. The API is the only way to read and write

to any of the GPIO pins on the Bolt hardware, which are all connected to the cloud. We have to create two applets (recipes) for this, one is to activate the alarm and the other is to de-activate the alarm. When the alarm is activated the capacitor connected to pin 3 should be charged and when the alarm is de-activated the capacitor connected to pin 4 should be charged.

For this service, the applet uses Google Assistant, and web hooks receive requests for it. The screenshot below demonstrates how the applet is configured to make the pin 3 high when the alarm is activated and the pin 4 high when the alarm is deactivated using Google Assistant.

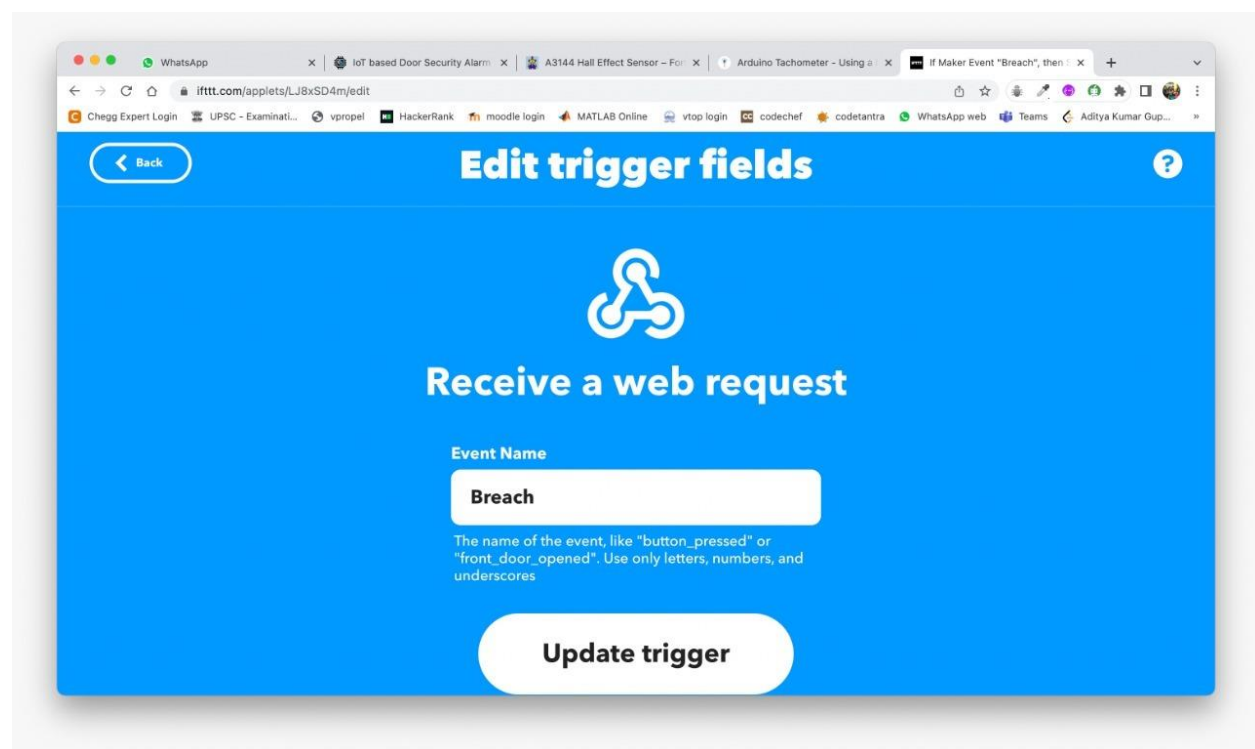


Figure 2: Applet

**Edit action fields**

**Send an email**

Email address  
edityakumargupta115@gmail.co ▼ Add new email

To address  
edityakumargupta115@gmail.com Add dependency

CC address  
ankitgha271@gmail.com Add dependency

BCC address  
Add dependency

Subject  
ALERT!! Add dependency

Body  
Security \* {EventName} \* on {OccurredAt} Add dependency

Attachment URL  
Add dependency

Update action

Figure 3: Applet

The two capacitors are used as memory components. Pins 3 and 4 serve as output pins when the board receives a command from IFTTT. The board will charge either Capacitor 1 or Capacitor 2 depending on the command.

Now that the IFTTT operation has been completed, the Python code has made pins 3 and 4 input pins. Based on which capacitor has charge, the Python code will determine whether to activate or deactivate the alarm by reading the status of these pins.



WhatsApp | IoT based Door Security Alarm | A3144 Hall Effect Sensor - For | Arduino Tachometer - Using a | If you say "Okay Google, Deac...

ifttt.com/applets/DUN9Click/edit

Chegg Expert Login | UPSC - Examinati... | vpropel | HackerRank | moodle login | MATLAB Online | vtop login | codechef | codetrantra | WhatsApp web | Teams | Aditya Kumar Gup...

Back

# Edit trigger fields

?

## Activate scene

Scene name

Deactivate office security system

Keep it short and simple. Don't include "Ok, Google" or "Activate"

Update trigger

Figure 4: Applet

WhatsApp | IoT based Door Security Alarm | A3144 Hall Effect Sensor - For | Arduino Tachometer - Using a | If you say "Okay Google, Deac...

ifttt.com/applets/DUN9Click/edit

Chegg Expert Login | UPSC - Examinati... | vpropel | HackerRank | moodle login | MATLAB Online | vtop login | codechef | codetrantra | WhatsApp web | Teams | Aditya Kumar Gup...

Back

# Edit action fields

?

## Make a web request

URL

https://cloud.boltiot.com/remote/20c9a837-ed50-487a-uc2f-1154-9b33235-digitalWrite?pin=&state=HIGH&deviceName=SOLT2910606

Method

GET

Content Type

application/json

Additional Headers

Body

Update action

Figure 5: Applet

When the intruder is found, the third applet will send an email. IFTTT's web hooks and Gmail service can be used for this. We will receive a URL from the Webhooks service that, when loaded, will send the pre-defined email. Below is a screenshot of the applet. The receiver ID and text can be changed however you like. If an intrusion is discovered, the requests library in the Python script will then trigger this URL.

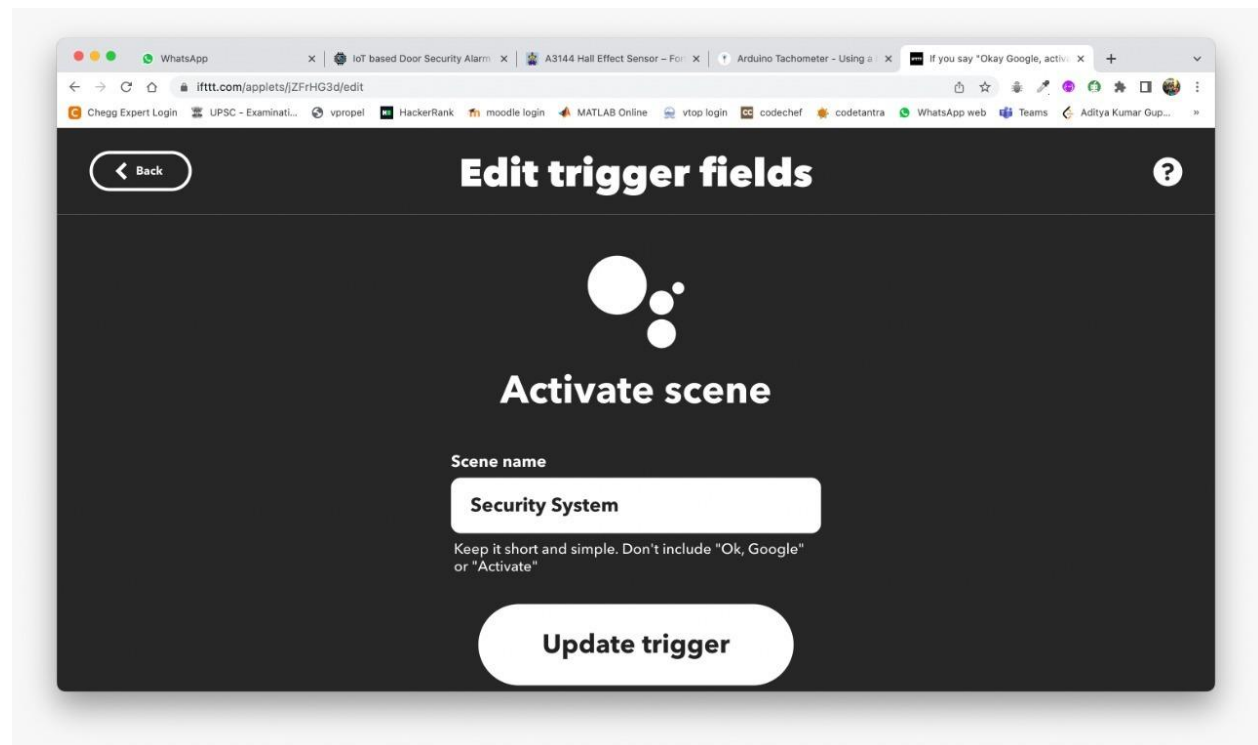


Figure 6: Applet

WhatsApp | IoT based Door Security Alarm | A3144 Hall Effect Sensor - For | Arduino Tachometer - Using a | if you say "Okay Google, activ...

ifttt.com/applets/jZFrHG3d/edit

Chegg Expert Login | UPSC - Examinati... | vpropel | HackerRank | moodle login | MATLAB Online | vtop login | codechef | codetantra | WhatsApp web | Teams | Aditya Kumar Gup...

### Edit action fields

Make a web request

URL

`https://cloud.bulbint.com/remote/2c09a337-ed50-4874-ac28-154c09ba3238/digitalWrite?pin=3&state=HIGH&deviceName=BOLT2910606`

Method

GET

Content type

application/json

Additional Headers

Body

Update action

Figure 7: Applet

## FLOW DIAGRAM/FLOW CHART: -

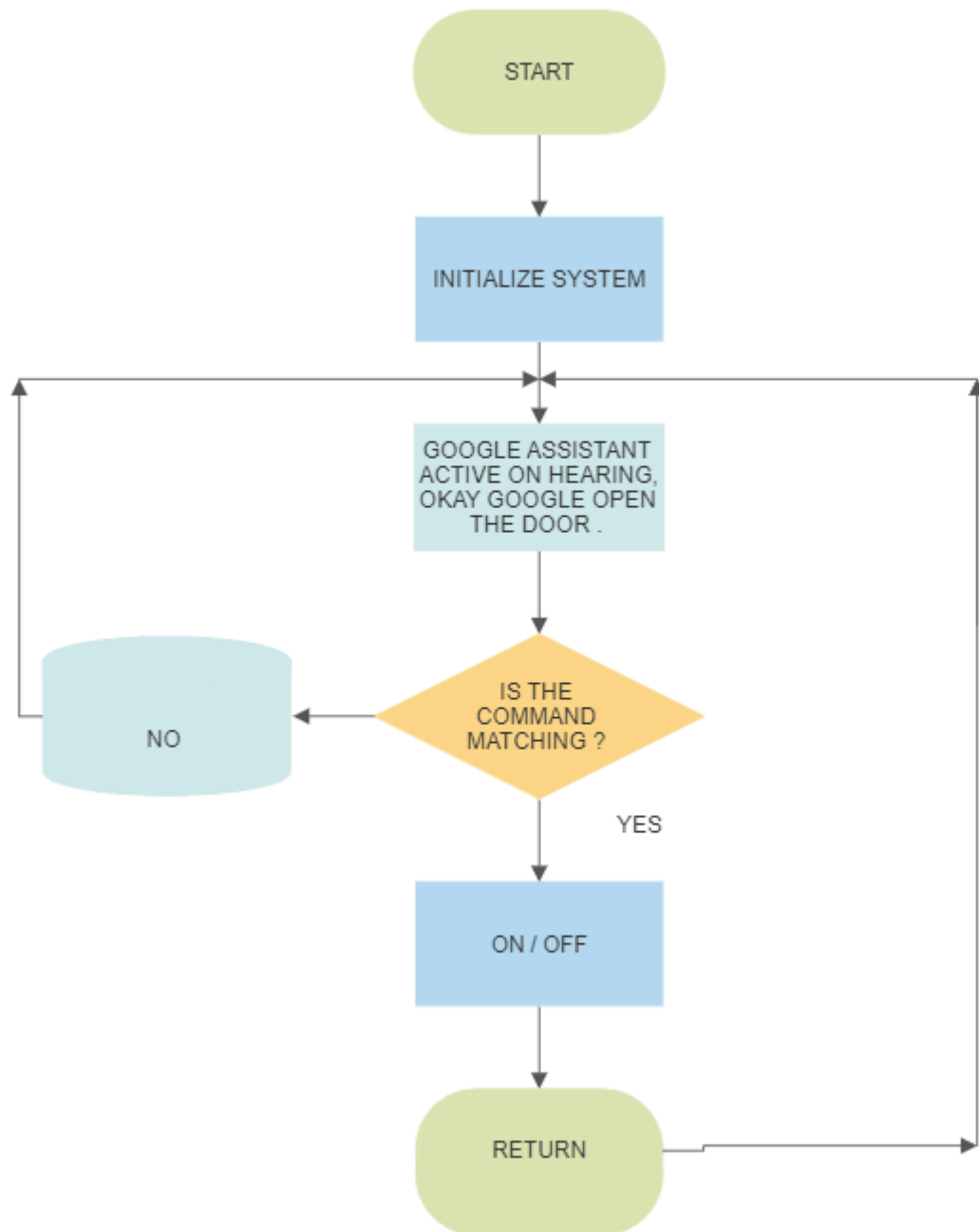


Figure 8: Flowchart

## CODE: -

```
import time #Import time to perform delay operations

import requests #use requests to send mail via webhooks IFTTT

from boltiot import Bolt #Import boltiot to control GPIO pins through API

api_key = "2c69a837-ed50-4874-ac2f-114c9fbb3328" #Get your API key from Bolt Cloud Website

device_id = "BOLT2910606" #Get your Bolt device ID from Bolt Cloud Website

mybolt = Bolt(api_key, device_id)

HIGH = '{"value": "1", "success": 1}' #This will be returned by bolt API if digital read is high

LOW = '{"value": "0", "success": 1}' #This will be returned by bolt API if digital read is low

alarm = 0 #Alarm is turned off by default

while True: #Infinite loop

    while alarm == 0: #If alarm is off

        response = mybolt.digitalRead('3') #check if it is being activated

        print('Alarm is activated? -', response)

        if (response == HIGH):

            print("Security System is activated")

            mybolt.digitalWrite('2', 'HIGH') #Turn on LED to indicate Alarm is activated

            alarm = 1

        elif (response == LOW):

            print ("Waiting for Security System to be activated....")

        else:

            print ("Problem in getting value from pin 3")

        time.sleep(15) #check once in every 15 seconds to avoid exceeding API rate limit

    while alarm == 1: #If alarm is on

        response = mybolt.digitalRead('4') #check if it is being de-activated

        print('Alarm is de-activated? -', response)
```

```

if (response == HIGH):

    print("Security System is De-activated")

    mybolt.digitalWrite('2', 'LOW')#Turn off LED to indicate Aalarm is De-activated

    alarm = 0

    time.sleep(15)

elif (response == LOW):

    print ("Security System is currently active can be deactivated from google assistant")

else:

    print ("Problem in getting value form pin 4")

response = mybolt.digitalRead('0') #check if hall sensor is triggered

if (response == HIGH): #if magnet is not present

    print ("Alert! Security breach Buzzer ON")

    mybolt.digitalWrite('1', 'HIGH')

    requests.get('https://maker.ifttt.com/trigger/Breach/json/with/key/ncc0s1wDOMwb63dn5u-ShK-3hCRHFs_0WVCfy5BaoTa') #webhook link to trigger mail through IFTTT

    requests.get('https://maker.ifttt.com/trigger/sms/json/with/key/ncc0s1wDOMwb63dn5u-ShK-3hCRHFs_0WVCfy5BaoTa') #webhook link to trigger SMS through IFTTT
    requests.get('https://maker.ifttt.com/trigger/SMS/json/with/key/f0zDZieIVBppq3hR0gTD7fvIWccfESiYFM2KM09dGKC')

    time.sleep(15)

    mybolt.digitalWrite('1', 'LOW')

    print ("Buzzer OFF")

elif (response == LOW):

    print ("No problem, all good!")

else:

    print ("Problem in reading the value of button")

time.sleep(15)

```

## OUTPUT SCREENSHOTS: -

The screenshot shows the VS Code editor with the `main.py` file open. The code is a Python script that interacts with a hardware device (likely a Raspberry Pi) to control a security system. The code includes comments in Hindi and English, and uses the `mybolt` library for hardware control.

```

29 print(response)
30
31 if (response == HIGH):
32
33     print("Security System is activated")
34
35     mybolt.digitalWrite('2', 'HIGH') #Turn on LED to indicate Aalarm is activa
36
37     alarm = 1
38
39 elif (response == LOW):
40
41     print ("Waiting for Security System to be activated....")
42
43 else:

```

The terminal output shows the execution of the script, including the activation of the security system and the LED, and the deactivation of the system and the LED.

```

Alert! Security breach Buzzer ON
Buzzer OFF
{"value": "1", "success": 1}
Security System is activated
Security System is De-activated
No problem, all good!
{"value": "Command timed out", "success": 0}
Problem in getting value form pin 3
{"value": "1", "success": 1}
Security System is activated
Security System is De-activated
Alert! Security breach Buzzer ON
Buzzer OFF
{"value": "1", "success": 1}
Security System is activated
Security System is De-activated

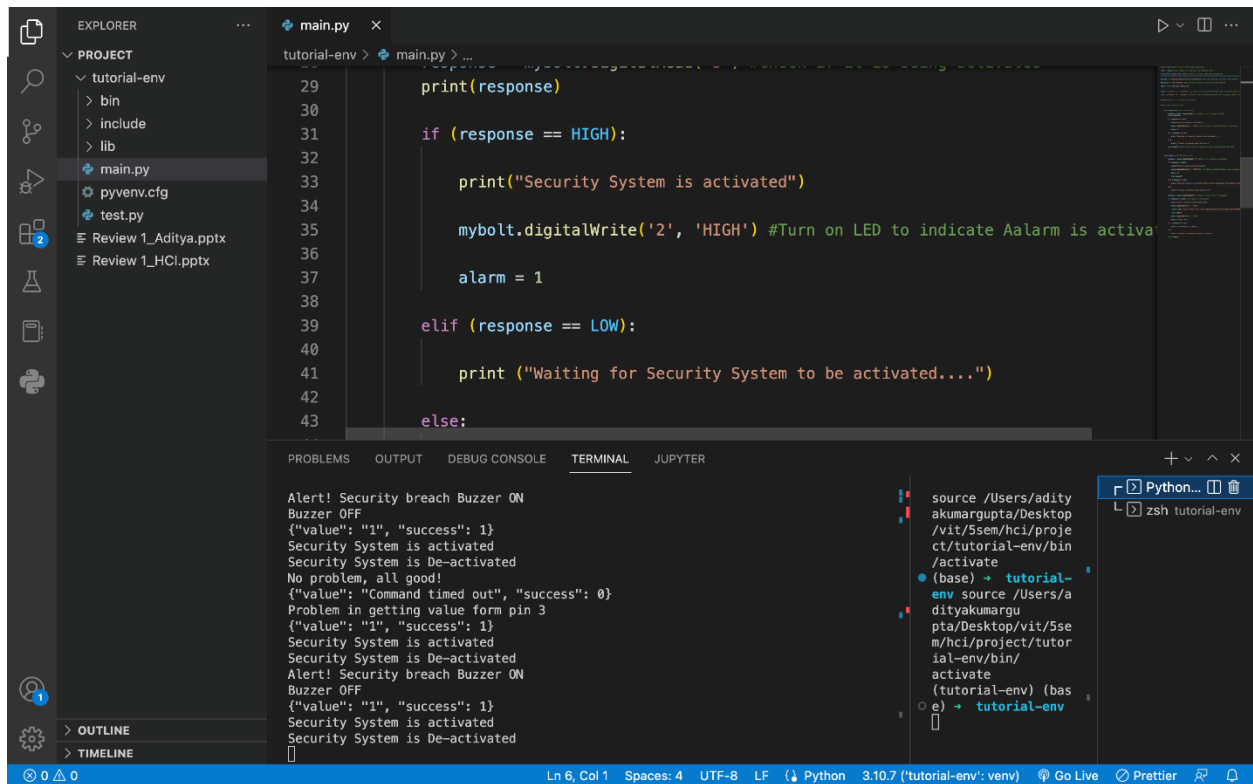
```

Figure 9: Flowchart

The screenshot shows the VS Code interface with the following details:

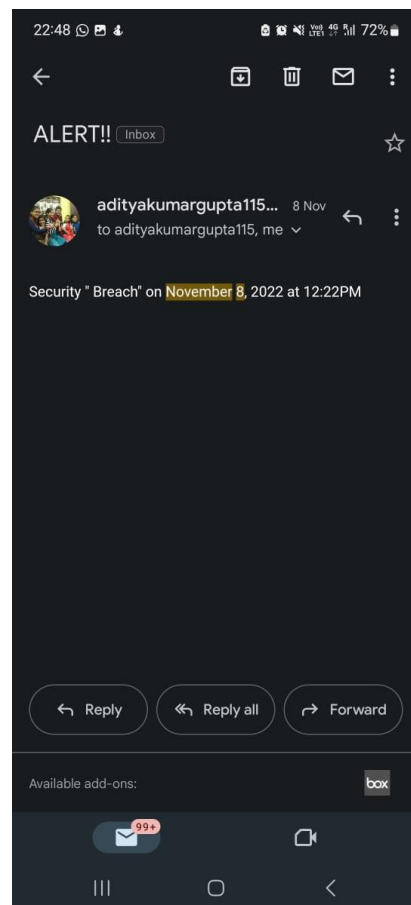
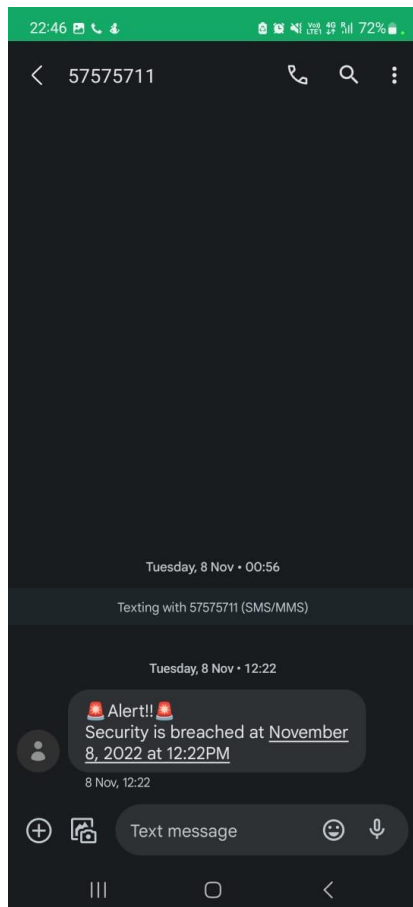
- Explorer Sidebar:** Contains files `main.py`, `pyvenv.cfg`, and `test.py`. Below them are two review items: `Review 1_Aditya.pptx` and `Review 1_HCI.pptx`.
- Run and Debug Console:** Shows the command `(tutorial-env) base ➤ tutorial-env python3 main.py` being executed. The output consists of multiple iterations of a simulated alarm system's behavior.
- Terminal Output:**

```
Alarm is activated? - {"value": "0", "success": 1}
Waiting for Security System to be activated...
Alarm is activated? - {"value": "0", "success": 1}
Waiting for Security System to be activated...
Alarm is activated? - {"value": "1", "success": 1}
Security System is activated
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
Alert! Security breach Buzzer ON
Buzzer OFF
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
Alarm is de-activated? - {"value": "0", "success": 1}
Security System is currently active can be deactivated from google assistant
No problem, all good!
```
- Status Bar:** At the bottom, it indicates the current file is `Ln 102, Col 22` in `Spaces: 4` encoding (`UTF-8`) with `LF` line endings. It also shows the Python version as `Python 3.10.7 (tutorial-env: venv)` and other extensions like Go Live and Prettier.



```
main.py
29 print(response)
30
31 if (response == HIGH):
32     print("Security System is activated")
33
34     mybolt.digitalWrite('2', 'HIGH') #Turn on LED to indicate Aalarm is activa
35
36     alarm = 1
37
38
39 elif (response == LOW):
40
41     print ("Waiting for Security System to be activated...")
42
43 else:
```

Alert! Security breach Buzzer ON  
Buzzer OFF  
{ "value": "1", "success": 1 }  
Security System is activated  
Security System is De-activated  
No problem, all good!  
{ "value": "Command timed out", "success": 0 }  
Problem in getting value form pin 3  
{ "value": "1", "success": 1 }  
Security System is activated  
Security System is De-activated  
Alert! Security breach Buzzer ON  
Buzzer OFF  
{ "value": "1", "success": 1 }  
Security System is activated  
Security System is De-activated





## ***PICTURES:-***



Fig. This figure represents coding in python and initializing the bolt program for enabling security system in cloud.

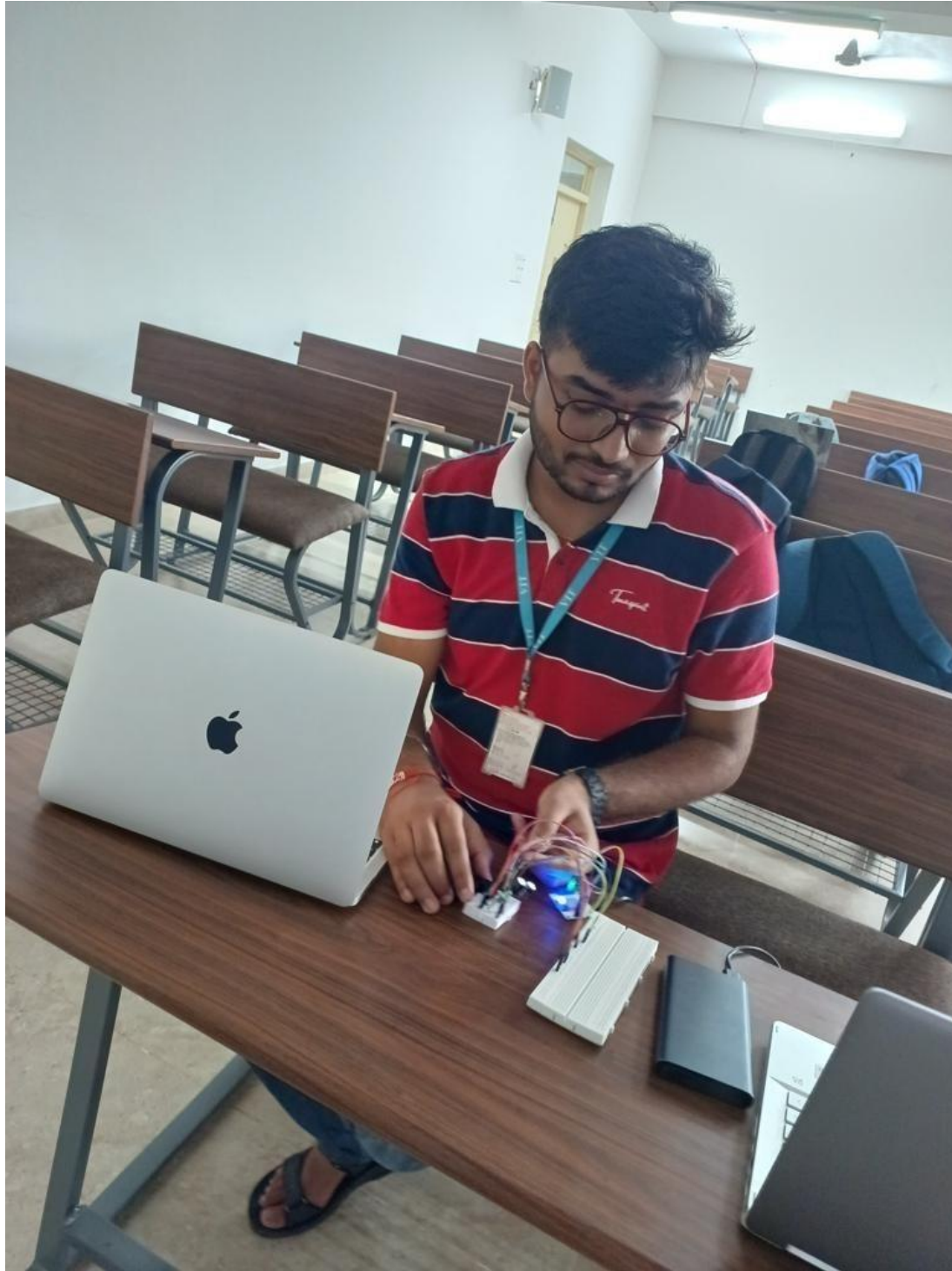


Fig. This figure represents activating the security system using google voice assistant and then breaching the security as removing the magnet.



Fig. This figure represents checking the status of the system after activating and deactivating it. Further cross checking it is using IFTTT server, whether every pipelines worked well.

## **CONCLUSION: -**

An interactive IoT-enabled home door security system with an app interface was designed and implemented in this project. Remotely, the user can keep an eye on and manage who has access to the front door. Any unauthorised attempts can be detected by the system, which can then raise an alarm from both the administrator's device and the security system itself. Compared to conventional locking security systems, this technique offers an additional level of protection and control. In conclusion, connectivity will be dominated by the Internet of Things. This makes it simple to connect a number of the everyday objects we use to the internet, enabling remote control of those objects without being physically present. Webcam and fingerprint integration are among the upcoming improvements. Instead of the user troubleshooting the issue, IoT also enables remote issue detection with the aid of built-in software and sensor-based fault detection systems. This has a strong chance of enhancing businesses, fine-tuning services and products, and better utilising resources. We firmly believe that our project is secure, hassle-free, and convenient for the end user and that the system proposed is workable because we are moving toward a future where all devices are connected to the cloud and to one another. Self-planning systems are expanding the savvy sector in the

modern technologically advanced world, ensuring that progress in the most recent development is consistently and quickly made on a variety of most recent adjusted portal lock security systems. As security becomes a priority or fundamental concern for everyone, the need for an advanced propelled passage lock security structure arises almost immediately. In light of the ongoing motivation to worry about this security and everything else, modified security structures are here to guide it. This project makes an effort to consider all universal door lock security systems in a comprehensive manner.

## **REFERENCES: -**

1. <https://circuitdigest.com/microcontroller-projects/iot-based-door-security-alarm-controlled-by-google-assistant>
2. <https://www.ijert.org/iot-based-smart-home-security-system-with-alert-and-door-access-control-using-smart-phone>
3. <https://developers.google.com/assistant/smarthome/guides/securitysystem>
4. [https://www.researchgate.net/publication/312559421\\_IoT\\_based\\_smart\\_security\\_and\\_home\\_automation\\_system](https://www.researchgate.net/publication/312559421_IoT_based_smart_security_and_home_automation_system)